20854

Network Working Group
NIC #20854
RFC #599

Robert T. Braden
UCLA/CCN
December 13, 1973

Postel

UPDATE ON NETRJS

A. INTRODUCTION

In July 1971, CCN published RFC #189 defining NETRJS, a private protocol for remote job entry. NETRJS provides a Network interface to CCN's rje program called RJS (Remote Job Service).(3) As noted in an earlier RFC,(6) "RJS" is the proper name of a software package existing only at CCN, not a generic term for rje.

For over two years now, CCN has provided rje service to the Network using NETRJS. We know of the following distinct implementations of NETRJS user programs:

RAND	OS/MVT on 370/158 (orginally on 360/65)
UCLA-NMC	SEX on Sigma 7
Illinois	ANTS on PDP-11
Utah	Tenex on PDP-10
MIT-DMCG	ITS on PDP-10
Harvard	DEC system on PDP-10
UCSB	OS/MVT on 360/75
ISI, BBN, NIC, I4	Tenex on PDP-10

We apologize to anyone slighted by omission from this list. Writing a new user process for NETRJS has proved to be a modest and straightforward task.

During the month of October, 1973, CCN processed 1373 batch jobs via NETRJS. The complete statistics are:

1,373	Jobs submitted
1,105	Jobs "printed"
0	Jobs "punched"
49,400	Cards "read"
822,900	Lines "printed"
18,907	Pages "printed"
393.6	Connect hours

The average job submitted was 360 lines ("cards"), and returned 745 lines on 17.1 pages. These figures are fairly typical.

Received at NIC 1-FEB-74

B. NEW ICP SOCKETS

At the request of the Socket Czar, Jon Postel, (see RFC #433) we intend to move the NETRJS ICP sockets from 11, 13, and 15 to 71, 73, and 75, respectively. At present, NETRJS is available from either socket subspace, so system programmers responsible for maintaining NETRJS user processes can switch over at their leisure. We plan to "decommit" sockets 11, 13, and 15 on July 1, 1974.

Those hosts which access NETRJS via socket 1 are unaffected.

C. NEW NETRJS

Last Fall, CCN installed a new implementation of its NETRJS server. An internal NETRJS rewrite was necessitated by other system changes and was timed to coincide with installation on September 5 of the "last release" of OS/360, Release 21.7. The new version of NETRJS contains a number of internal improvements over the orginal version written two years ago. There are also a few external differences, as follows:

1. No More Squish

The long-standing "squish" problem in NETRJS has been fixed. This problem arose because of the "squishiness" of Network data transfer , i.e. the variable delay between originator and receiver processes due to NCP buffering. The result was that a short print output file could be "transmitted" by RJS, dequeued, and discarded at CCN before the first message had actually reached the remote host. If the remote host crashed or the user tried to cancel (and save) the output stream, it was too late; the output was lost in the "squish". We were careless about this in the first version. Now NETRJS awaits the RFNM from the end-of-data mark before telling RJS to discard the job output.

2. Timeouts

The new version is a little tougher on timeouts, to free CCN resources when users are slow.

a. Signon Timeout

If the user, after connecting to NETRJS and receiving the READY message, fails to send a valid SIGNON command within 3 minutes, CCN will close the Telnet connections.

b. Data Transfer Timeout

(1) CCN will abort the READER data transfer connection if the user site leaves the connection open without sending any bits for 5 minutes.

(2) CCN will abort the PRINTER or PUNCH data transfer connection if the user site stops accepting bits for 5 minutes.

3. New Messages

The NETRJS messages to the remote terminal have been revised to better distinguish problems at CCN, at the user site, or in the Network. See Reference 8 for a complete list.

4. Subsystem Interrupt

The user can send a Control-C to terminate his NETRJS session either before or after signon. Continuation is not possible after the Control-C.

This provides an escape for a user who for some reason can't signon or signoff or close his Telnet connection. If the user entered via the RJS command in Socket 1, Control C will return him to the Server Telnet command level.

One other improvement will reduce user frustration: NETRJS now returns an INVALID SIGNON message if the user enters anything but a valid SIGNON command after initially connecting to the NETRJS server.

D. CLARIFICATIONS AND AMENDMENTS TO NETRJS PROTOCOL

Over the past two years, system programmers writing NETRJS user processes have pointed out areas of the protocol which were poorly defined in RFC #189. In addition a few minor changes have been made, largely as the result of implementation accidents.

1. The jobname header of a print file does not have an ASA carriage control byte. However, it will be encoded in the format (compressed or truncated) selected by a particular VRBT.

2. The punch connection sends 81 byte records, the first byte being a blank carriage control character. This is contrary to RFC #189 and is illogical; it was an implementation bug which we kept for compatibility.

3. Page 3 of RFC #189 defined fixed values for the user's data transfer sockets relative to his Telnet sockets. In fact, NETRJS does not enforce these user data transfer sockets but will accept RFC's for any user sockets.

4. RFC #189 specified a choice of two character mappings for the virtual remote batch terminal: EBCDIC and ASCII (-68). An ASCII-63 mapping was later added for the convenience of users with Model 33-like keyboards (RAND, actually). The ASCII-63 mapping is selected by doing an ICP to socket 75 or by entering "TTYRJS" in CCN's Telnet Server. Figure 1 shows the actual ASCII-63 mapping in use today. This supercedes the earlier version of this mapping, shown in RFC 338.

5. The ASCII-68 mapping specified in RFC 189 was also changed to provide unique mappings for all ASCII characters. The present ASCII-68 mapping used by both NETRJS and TSO at CCN is shown in Figure 1.

E. RJS TERMINAL OPTIONS

When a new NETRJS virtual terminal is defined, certain options are available; these options are listed below. If the user does not specify otherwise, CCN will use truncated data format and turn all other options on.

1. Truncated /Compressed Data Format

As explained in RFC 189, a virtual remote batch terminal under RJS may use either the truncated data format (default) or the compressed format for printer and punch output. With the truncated format, CCN merely removes trailing blanks from each output line; if compressed format is specified, CCN will also encode strings of imbedded blanks or other repeated characters. CCN will accept either format in the card reader stream, regardless of the terminal option. See Reference 9 for discussion of the virtues of compression.

2. Automatic Coldstart Job Resubmission

If "R" (Restart) is specified in the accounting field on the JOB card and if this option is chosen, RJS will automatically resubmit the job from the beginning if the CCN operating system should be "coldstarted" before all output from the job is returned. Otherwise, the job will be lost and must be resubmitted from the remote terminal in case of a coldstart.

3. Automatic Output RESTART

With this option, transmission of printer output which is interrupted by a broken connection always starts over at the beginning. Without this option, the output is backspaced approximately one page when restarted, unless the user forces the output to start over from the beginning with a RESTART command when the printer connection is re-opened and before printing begins.

4. Password Protection

This option allows a password to be supplied when a terminal is signed on, preventing unauthorized use of the terminal ID.

5. Suppression of Punch Separator and Large Letters.

This option suppresses both separator cards which RJS normally puts in front of each punched output deck, and separator pages on printed output containing the job name in large block letters. These separators are an operational aid when the output is directed to a real printer or punch, but generally undesirable for an ARPA user who is saving the output in a file for on-line examination.

F. WARNING ON TENEX NETRJS USER PROCESS (6)

The Tenex implementation of a NETRJS user program is a command normally called "RJS". This program has some pitfalls of which users should be aware.

1. For strictly historical reasons, the commonly-available version of the Tenex RJS command uses Socket 15, and therefore, the ASCII-63 translation. We hope to propagate soon a version which uses the ASCII-68 mapping via Socket 73, and stamp out the earlier version.

2. The Tenex RJS command fails to recognize the US character sometimes used instead of CR LF as end-of-line. As noted in RFC 571, the Tenex user FTP program has the same problem.

3. The Tenex RJS command truncates without warning card images exceeding 80 characters in length.

G. REFERENCES ON NETRJS

1. "Interim NETRJS Specifications", R. T. Braden. RFC #189: NIC #7133, July 15, 1971.

   This is the basic system programmer's definition document, and is really the final specification. The proposed changes mentioned on the first page of RFC #189 were never implemented, since the DTP then in vogue became obsolete.

2. "NETRJS Remote Operator Commands", R. T. Braden. NIC #7182, August 9, 1971

   This document together with References 3 and 8 define the remote operator (i.e. user) command language for NETRJS, and form the basic user documentation for NETRJS at CCN.

3. "Implementation of a Remote Job Service", V. Martin and T. W. Springer. NIC #7183, July, 1971.

4. "Remote Job Entry to CCN via UCLA Sigma 7; A Scenario", UCLA/CCN. NIC #7748, November 15, 1971.

   This document described the first NETRJS user implementation available on a server host. Thisprogram is no longer of general interest.

5. "Using Network Remote Job Entry", E. F. Harslem. RFC #307: NIC #9258, February 24, 1972.

   This document is out of date, but describes generally the Tenex NETRJS user process "RJS".

6. "EBCDIC/ASCII Mapping for Network RJS", R. T. Braden. RFC #338: NIC #9931, May 17, 1972.

The ASCII-63 mapping described here is no longer correct, but CCN's standard ASCII-68/EBCDIC mapping is described correctly.

7. "NETRJT--Remote Job Service Protocol for TIP's", R. T. Braden. RFC #283: NIC #8165, December 20, 1971.

This was an attempt to define an rje protocol to handle TIPs. Although NETRJT was never implemented, many of its features are incorporated in the current Network standard RJE protocol.

8. "CCN NETRJS Server Messages to Remote User", R. T. Braden. NIC #20268, November 26, 1973.

9. "FTP Data Compression", R. T. Braden. RFC #468: NIC #14742, March 8, 1973.

FIGURE 1. NETRJS CHARACTER MAPPINGS AT UCLA-CCN

The character set of the VRBT (Virtual Remote Batch Terminal) is determined by the initial connection to RJS, as follows:

VRBT Character Set	ICP Socket	OR	Server Telnet Command
EBCDIC	71		RJS
ASCII-68	73		ARJS
ASCII-63(tty)	75		TTYRJS

These mappings are as follows:

ASCII-68 Mapping:

Corresponding graphics are mapped one-to-one.

Unmatched graphics are mapped as in the table below.

ASCII-68 controls are mapped one-to-one onto the matching EBCDIC controls, with DC4(ASCII) mapped onto TM(EBCDIC).

ASCII-63 Mapping:

Corresponding graphics are mapped one-to-one.

ASCII codes X'61' - X'7A' (the ASCII-68 lower case letters) are mapped onto EBCDIC lower case.

Unmatched graphics are mapped as shown in the table below.

ASCII-63 controls X'00' - X'1F' are mapped as for ASCII-68.

ASCII codes X'60' and X'7B' - X'7E' are mapped as shown in the following table.

EBCDIC		ASCII-68 VRBT		ASCII-63 VRBT	
¦	X'4F'	¦	X'7C'	[	X'5B'
¬	X'5F'	~	X'7E'	]	X'5D'
¢	X'4A'	\	X'5C'	\	X'5C'
underscore	X'6D'		X'5F'	left-arrow	X'5F'
	X'71'	^	X'5E'	up-arrow	X'5E'
([)	X'AD'	[	X'5B'		X'7C'
(])	X'BD'	]	X'5D'		X'7E'
	X'8B'	{	X'7B'		X'7B'
	X'9B'	}	X'7D'		X'7D'
	X'79'	`	X'60'		X'60'

Note: this page is available on-line as HELP RJSCHARS in CCN's Telnet Server (Socket 1). The on-line version is set up to be typed out on an ASCII-68 terminal.

Network Working Group
NIC #20854
RFC #599

Robert T. Braden
UCLA/CCN
December 13, 1973

UPDATE ON NETRJS

A. INTRODUCTION

In July 1971, CCN published RFC #189 defining NETRJS, a private protocol for remote job entry. NETRJS provides a Network interface to CCN's rje program called RJS (Remote Job Service).[3] As noted in an earlier RFC,[6] "RJS" is the proper name of a software package existing only at CCN, not a generic term for rje.

For over two years now, CCN has provided rje service to the Network using NETRJS. We know of the following distinct implementations of NETRJS user programs:

RAND	OS/MVT on 370/158 (orginally on 360/65)
UCLA-NMC	SEX on Sigma 7
Illinois	ANTS on PDP-11
Utah	Tenex on PDP-10
MIT-DMCG	ITS on PDP-10
Harvard	DEC system on PDP-10
UCSB	OS/MVT on 360/75
ISI, BBN, NIC, I4	Tenex on PDP-10

We apologize to anyone slighted by omission from this list. Writing a new user process for NETRJS has proved to be a modest and straightforward task.

During the month of October, 1973, CCN processed 1373 batch jobs via NETRJS. The complete statistics are:

1,373	Jobs submitted
1,105	Jobs "printed"
0	Jobs "punched"
49,400	Cards "read"
822,900	Lines "printed"
18,907	Pages "printed"
393.6	Connect hours

The average job submitted was 360 lines ("cards"), and returned 745 lines on 17.1 pages. These figures are fairly typical.

B. NEW ICP SOCKETS

At the request of the Socket Czar, Jon Postel, (see RFC #433) we intend to move the NETRJS ICP sockets from 11, 13, and 15 to 71, 73, and 75, respectively. At present, NETRJS is available from either socket subspace, so system programmers responsible for maintaining NETRJS user processes can switch over at their leisure. We plan to "decommit" sockets 11, 13, and 15 on July 1, 1974.

Those hosts which access NETRJS via socket 1 are unaffected.

C. NEW NETRJS

Last Fall, CCN installed a new implementation of its NETRJS server. An internal NETRJS rewrite was necessitated by other system changes and was timed to coincide with installation on September 5 of the "last release" of OS/360, Release 21.7. The new version of NETRJS contains a number of internal improvements over the orginal version written two years ago. There are also a few external differences, as follows:

1. No More Squish

The long-standing "squish" problem in NETRJS has been fixed. This problem arose because of the "squishiness" of Network data transfer , i.e. the variable delay between originator and receiver processes due to NCP buffering. The result was that a short print output file could be "transmitted" by RJS, dequeued, and discarded at CCN before the first message had actually reached the remote host. If the remote host crashed or the user tried to cancel (and save) the output stream, it was too late; the output was lost in the "squish". We were careless about this in the first version. Now NETRJS awaits the RFNM from the end-of-data mark before telling RJS to discard the job output.

2. Timeouts

The new version is a little tougher on timeouts, to free CCN resources when users are slow.

a. Signon Timeout

If the user, after connecting to NETRJS and receiving the READY message, fails to send a valid SIGNON command within 3 minutes, CCN will close the Telnet connections.

b. Data Transfer Timeout

(1) CCN will abort the READER data transfer connection if the user site leaves the connection open without sending any bits for 5 minutes.

(2) CCN will abort the PRINTER or PUNCH data transfer connection if the user site stops accepting bits for 5 minutes.

3. New Messages

The NETRJS messages to the remote terminal have been revised to better distinguish problems at CCN, at the user site, or in the Network. See Reference 8 for a complete list.

4. Subsystem Interrupt

The user can send a Control-C to terminate his NETRJS session either before or after signon. Continuation is not possible after the Control-C.

This provides an escape for a user who for some reason can't signon or signoff or close his Telnet connection. If the user entered via the RJS command in Socket 1, Control C will return him to the Server Telnet command level.

One other improvement will reduce user frustration: NETRJS now returns an INVALID SIGNON message if the user enters anything but a valid SIGNON command after initially connecting to the NETRJS server.

D. CLARIFICATIONS AND AMENDMENTS TO NETRJS PROTOCOL

Over the past two years, system programmers writing NETRJS user processes have pointed out areas of the protocol which were poorly defined in RFC #189. In addition a few minor changes have been made, largely as the result of implementation accidents.

1. The jobname header of a print file does not have an ASA carriage control byte. However, it will be encoded in the format (compressed or truncated) selected by a particular VRBT.

2. The punch connection sends 81 byte records, the first byte being a blank carriage control character. This is contrary to RFC #189 and is illogical; it was an implementation bug which we kept for compatibility.

3. Page 3 of RFC #189 defined fixed values for the user's data transfer sockets relative to his Telnet sockets. In fact, NETRJS does not enforce these user data transfer sockets but will accept RFC's for any user sockets.

(2) CCN will abort the PRINTER or PUNCH data transfer connection if the user site stops accepting bits for 5 minutes.

3. New Messages

The NETBLT messages in the remote terminal have been revised to better distinguish problems at CCN, at the user site, or in the Network. See Reference 5 for a complete list.

4. Subsystem Interrupt

The user can send a Control-C to terminate his NETBLT session either before or after signon. Continuation is not possible after the Control-C.

This provides an escape for a user who for some reason can't signon or signoff or clear his Telnet connection. If the user entered via the GLS command in Socket 1, Control C will return him to the Server Telnet command level.

One other improvement will reduce user frustration. NETBLT now returns an INVALID SIGNON message if the user enters anything but a valid SIGNON command after initially connecting to the NETBLT server.

6. CLARIFICATIONS AND ANCHOWERTS TO NETBLT PROTOCOL

Over the past two years, system programmers writing NETBLT user protocols have pinpointed our areas of the protocol which were poorly defined in RFC 969. In addition a few minor changes have been made, largely as the result of implementation activity.

1. The Jumbogram header of a point file does not have an ASA carriage control byte. However, it will be encoded in the format (compressed or truncated) selected by a particular VRST.

2. The quick connection packet at byte rebuild, the first byte being a blank carriage control character. This is contrary to RFC 969 and is illogical; it was an implementation bug which we hope for compatibility.

3. Page 1 of RFC 969 defines fixed values for the user's data transfer sockets relative to his Telnet socket. In fact, NETBLT does not enforce these user data transfer sockets but will accept BOTH for any user socket.

4. RFC #189 specified a choice of two character mappings for the virtual remote batch terminal: EBCDIC and ASCII (-68). An ASCII-63 mapping was later added for the convenience of users with Model 33-like keyboards (RAND, actually). The ASCII-63 mapping is selected by doing an ICP to socket 75 or by entering "TTYRJS" in CCN's Telnet Server. Figure 1 shows the actual ASCII-63 mapping in use today. This supercedes the earlier version of this mapping, shown in RFC 338.

5. The ASCII-68 mapping specified in RFC 189 was also changed to provide unique mappings for all ASCII characters. The present ASCII-68 mapping used by both NETRJS and TSO at CCN is shown in Figure 1.

E. RJS TERMINAL OPTIONS

When a new NETRJS virtual terminal is defined, certain options are available; these options are listed below. If the user does not specify otherwise, CCN will use truncated data format and turn all other options on.

1. Truncated /Compressed Data Format

As explained in RFC 189, a virtual remote batch terminal under RJS may use either the truncated data format (default) or the compressed format for printer and punch output. With the truncated format, CCN merely removes trailing blanks from each output line; if compressed format is specified, CCN will also encode strings of imbedded blanks or other repeated characters. CCN will accept either format in the card reader stream, regardless of the terminal option. See Reference 9 for discussion of the virtues of compression.

2. Automatic Coldstart Job Resubmission

If "R" (Restart) is specified in the accounting field on the JOB card and if this option is chosen, RJS will automatically resubmit the job from the beginning if the CCN operating system should be "coldstarted" before all output from the job is returned. Otherwise, the job will be lost and must be resubmitted from the remote terminal in case of a coldstart.

3. Automatic Output RESTART

With this option, transmission of printer output which is interrupted by a broken connection always starts over at the beginning. Without this option, the output is backspaced approximately one page when restarted, unless the user forces the output to start over from the beginning with a RESTART command when the printer connection is re-opened and before printing begins.

4. RFC 1489 specified a choice of two character mappings for the virtual remote batch terminal: EBCDIC and ASCII (-68). An ASCII-63 mapping was later added for the convenience of users with Model 33-ASCII keyboards (TWO, actually). The ASCII-63 mapping is selected by doing an RCV to socket 75 or by entering "TTMAS" in CCN's Telnet server. Figure 1 shows the actual ASCII-63 mapping in use today. This supersedes the earlier version of this mapping, shown in RFC 324.

5. The ASCII-68 mapping specified in RFC 180 was also changed to provide unique mappings for all ASCII characters. The printer ASCII-68 mapping used by both METAS and TSO at CCN is shown in Figure 1.

E. RJS TERMINAL OPTIONS

When a new METAS virtual terminal is defined, certain options are available; these options are listed below. If the user does not specify otherwise, CCN will use translated data format and turn all other options on.

1. Translated/Compressed Data Format

As explained in RFC 180, a virtual remote batch terminal under RJS may use either the translated data format (default) or the compressed format for printer and punch output. With the translated format, CCN merely removes trailing blanks from each output line; if compressed format is requested, CCN will also encode strings of embedded blanks or other repeated characters. CCN will accept either format at the same combo stream, regardless of the terminal option. See Reference 2 for the explanation of the various options.

2. Automatic Collatert Job Abandonment

If "N" (default) is specified in the accounting field on the JOB card and if this option is chosen, RJS will automatically resubmit the job from the beginning if the CCN operating system should be "collatertal" before all output from the job is returned. Otherwise, the job will be lost and must be resubmitted from the remote terminal in case of a collatert.

3. Automatic Output RESTART

With this option, transmission of printer output which is interrupted by a broken connection always starts over at the beginning. Without this option, the output is backspaced approximately one page when restarted, unless the user forces the output to start over from the beginning with a RESTART command when the printer connection is re-opened and before printing begins.

4. Password Protection

This option allows a password to be supplied when a terminal is signed on, preventing unauthorized use of the terminal ID.

5. Suppression of Punch Separator and Large Letters

This option suppresses both separator cards which RJS normally puts in front of each punched output deck, and separator pages on printed output containing the job name in large block letters. These separators are an operational aid when the output is directed to a real printer or punch, but generally undesirable for an ARPA user who is saving the output in a file for on-line examination.

F. WARNING ON TENEX NETRJS USER PROCESS (6)

The Tenex implementation of a NETRJS user program is a command normally called "RJS". This program has some pitfalls of which users should be aware.

1. For strictly historical reasons, the commonly-available version of the Tenex RJS command uses Socket 15, and therefore, the ASCII-63 translation. We hope to propagate soon a version which uses the ASCII-68 mapping via Socket 73, and stamp out the earlier version.

2. The Tenex RJS command fails to recognize the US character sometimes used instead of CR LF as end-of-line. As noted in RFC 571, the Tenex user FTP program has the same problem.

3. The Tenex RJS command truncates without warning card images exceeding 80 characters in length.

G. REFERENCES ON NETRJS

1. "Interim NETRJS Specifications", R. T. Braden. RFC #189: NIC #7133, July 15, 1971.

   This is the basic system programmer's definition document, and is really the final specification. The proposed changes mentioned on the first page of RFC #189 were never implemented, since the DTP then in vogue became obsolete.

2. "NETRJS Remote Operator Commands", R. T. Braden. NIC #7182, August 9, 1971

   This document together with References 3 and 8 define the remote operator (i.e. user) command language for NETRJS, and form the basic user documentation for NETRJS at CCN.

3. "Implementation of a Remote Job Service", V. Martin and T. W. Springer. NIC #7183, July, 1971.

4. "Remote Job Entry to CCN via UCLA Sigma 7; A Scenario", UCLA/CCN. NIC #7748, November 15, 1971.

   This document described the first NETRJS user implementation available on a server host. Thisprogram is no longer of general interest.

5. "Using Network Remote Job Entry", E. F. Harslem. RFC #307: NIC #9258, February 24, 1972.

   This document is out of date, but describes generally the Tenex NETRJS user process "RJS".

G. REFERENCES ON METALS

1. "Interim METALS Specifications", R. T. Braden. NIC #1243, July 15, 1971.

   This is the basic system programmer's definition document, and is really the final specification. The proposed changes mentioned on the first page of NIC #1259 were never implemented, since the OFF then in vogue became obsolete.

2. "METALS Remote Operator Commands", R. T. Braden. NIC #7182, August 9, 1971.

   This document together with References 3 and 4 defines the remote operator (i.e. user) command language for METALS, and forms the basic user documentation for METALS at UCSB.

3. "Implementation of a Remote Job Service", V. Mattin and T. R. Springer. NIC #7181, July, 1971.

4. "Remote Job Entry to UCM via UCLA Sigma 7: A Scenario", McKay/Ellis. NIC #7248, November 15, 1971.

   This document described the first METALS user implementation available on a server host. This document is no longer of general interest.

5. "Using Network Remote Job Entry", R. T. Braden. RFC #327, NIC #8239, February 24, 1972.

   This document is out of date, but describes generally the Tenex METALS user process "NJE".

6. "EBCDIC/ASCII Mapping for Network RJS", R. T. Braden. RFC #338: NIC #9931, May 17, 1972.

The ASCII-63 mapping described here is no longer correct, but CCN's standard ASCII-68/EBCDIC mapping is described correctly.

7. "NETRJT--Remote Job Service Protocol for TIP's", R. T. Braden. RFC #283: NIC #8165, December 20, 1971.

This was an attempt to define an rje protocol to handle TIPs. Although NETRJT was never implemented, many of its features are incorporated in the current Network standard RJE protocol.

8. "CCN NETRJS Server Messages to Remote User", R. T. Braden. NIC #20268, November 26, 1973.

9. "FTP Data Compression", R. T. Braden. RFC #468: NIC #14742, March 8, 1973.

FIGURE 1. NETRJS CHARACTER MAPPINGS AT UCLA-CCN

The character set of the VRBT (Virtual Remote Batch Terminal) is determined by the initial connection to RJS, as follows:

VRBT Character Set	ICP Socket	OR	Server Telnet Command
EBCDIC	71		RJS
ASCII-68	73		ARJS
ASCII-63(tty)	75		TTYRJS

These mappings are as follows:

ASCII-68 Mapping:

Corresponding graphics are mapped one-to-one.

Unmatched graphics are mapped as in the table below.

ASCII-68 controls are mapped one-to-one onto the matching EBCDIC controls, with DC4(ASCII) mapped onto TM(EBCDIC).

ASCII-63 Mapping:

Corresponding graphics are mapped one-to-one.

ASCII codes X'61' - X'7A' (the ASCII-68 lower case letters) are mapped onto EBCDIC lower case.

Unmatched graphics are mapped as shown in the table below.

ASCII-63 controls X'00' - X'1F' are mapped as for ASCII-68.

ASCII codes X'60' and X'7B' - X'7E' are mapped as shown in the following table.

EBCDIC		ASCII-68 VRBT		ASCII-63 VRBT	
¦	X'4F'	¦	X'7C'	[	X'5B'
¬	X'5F'	~	X'7E'	]	X'5D'
¢	X'4A'	\	X'5C'	\	X'5C'
underscore	X'6D'	_	X'5F'	left-arrow	X'5F'
	X'71'	^	X'5E'	up-arrow	X'5E'
(¢)	X'AD'	[	X'5B'		X'7C'
(])	X'BD'	]	X'5D'		X'7E'
	X'8B'	{	X'7B'		X'7B'
	X'9B'	}	X'7D'		X'7D'
	X'79'	`	X'60'		X'60'

Note: this page is available on-line as HELP RJSCHARS in CCN's Telnet Server (Socket 1). The on-line version is set up to be typed out on an ASCII-68 terminal.

The character set of the VMNT (Virtual Remote Batch Terminal) is determined by the initial connection to RJS, as follows:

VMNT Character Set	ICF Socket	OR	Server Telnet Command
EBCDIC	71		RJS
ASCII-68	73		AMJS
ASCII-63 (rsy)	75		STYLUS

These mappings are as follows:

ASCII-68 Mapping:

Corresponding graphics are mapped one-to-one.

Unmatched graphics are mapped as in the table below.

ASCII-68 controls are mapped one-to-one onto the matching EBCDIC controls, with DC4(ASCII) mapped onto TM(EBCDIC).

ASCII-63 Mapping:

Corresponding graphics are mapped one-to-one.

ASCII codes X'61' - X'7A' (the ASCII-68 lower case letters) are mapped onto EBCDIC lower case.

Unmatched graphics are mapped as shown in the table below.

ASCII-63 controls X'00' - X'1F' are mapped as for ASCII-68.

ASCII codes X'60' and X'7B' - X'7E' are mapped as shown in the following table.

EBCDIC	ASCII-68 VMNT	ASCII-63 VMNT
! X'4F'	! X'7C'	[X'5B'
¬ X'5F'	~ X'7E'	] X'5D'
¢ X'4A'	/ X'5C'	/ X'5C'
underscore X'6D'	X'5F'	left-arrow X'5F'
X'71'	^ X'5E'	up-arrow X'5E'
(C) X'AD'	{ X'5B'	X'7C'
(J) X'BD'	} X'5D'	X'7E'
X'8B'	{ X'7B'	X'7E'
X'9B'	} X'7D'	X'7D'
X'79'	` X'60'	X'60'

Note: this page is available on-line as MELP RJSCHARS in CCV's Telnet Server (Socket 1). The on-line version is set up to be typed out on an ASCII-63 terminal.